A ROUTER PROVIDING CONTINUITY OF SERVICE OF THE STATE MACHINES ASSOCIATED WITH THE NEIGHBORING ROUTERS

The present invention relates to continuity of the routing service in a network of the Internet Protocol (IP) type. More precisely, the invention can be applied particularly well to the Open Shortest Path First (OSPF) protocol, as defined in the Request For Comments (RFC) 2328 of the Internet Engineering Task Force (IETF).

BACKGROUND OF THE INVENTION

The OSPF protocol is a protocol of the Transmission Control Protocol/Internet Protocol (TCP/IP) family making it possible for the routing systems or "routers" of an Internet network to have sufficient knowledge of the network to route the received packets correctly to their destinations.

The particularity of the OSPF protocol is that it is a dynamic routing protocol, i.e. it is capable of taking account of changes in the topology of the network in dynamic manner. For this purpose, the protocol has steps in which messages are exchanged periodically, in order to update constantly the knowledge that each router possesses of the network or of a portion of the network.

A router has connections with other routers. These connections can be of various types, such as:

point to point networks;

multi-access networks, e.g. of the Ethernet™ type; and

stub networks making it possible to interconnect a set of host stations.

As a function of the type of connection and of the status of the router, it can be necessary to put the router into adjacency with another router.

Putting two routers into adjacency consists in ensuring that they share exactly the same data about the topology of the network.

In some cases, it is not necessary to put two routers into adjacency. For example, in a multi-access network, for optimization reasons, adjacency is implemented only between each router and a router that is elected to be the designated router.

In the OSPF protocol, the designated router R_1 exchanges various types of messages with the neighboring routers, such as Database Description Packet (DDP) messages, "Hello" messages, data transmission (Link State (LS) Update) messages, data request (LS Request) messages, and acknowledgement of receipt (LS Acknowledgement) messages.

The object of the "Hello" messages is to inform the other routers periodically that the issuer of the message is still active.

The LS Update messages make it possible to receive data about the routers making up the network, while the LS Request messages make it possible to request data about the routers.

In compliance with the OSPF protocol, each router has a routing table that makes it possible to forward correctly the messages that it receives. Because of the dynamic aspect of the network, the routing tables must be constantly updated.

These updates are effected in particular by exchanging messages containing fragmentary information about the network and referred to as Link State Advertisement (LSA) messages. The routing tables are calculated by each router on the basis of this data.

The LS Update messages are in fact collections of LSA fragmentary data.

A third type of message is the Database Description Packet (DDP) message. These messages make it possible for two routers to exchange LSA summary lists, i.e. descriptions of the contents of their databases.

The various messages are described more fully in above-mentioned RFC 2328.

In the OSPF protocol, provision is also made to associate a state machine with each router that is a neighbor of a router.

Figure 1 represents such a state machine. As is conventional, the circles represent the states which the interface can occupy. At any given time, the interface must be in one of these states. Each arrow in the diagram represents a transition from one state to another. The names of the states are given in the diagram, as they appear in RFC 2328.

The initial state of the state machine is represented by the circle referenced "Down".

When the router receives a "Hello" message, the "Hello Received" event is generated and the state machine goes to an "Init" state. The "Start" event is generated when a "Hello" message must be generated firstly by the router, in a non-broadcast multiple access network. In which case, the state machine goes to the "Attempt" state. A "Hello" message being received in response then generates a "Hello Received" event, and the state machine goes to the "Init" state.

As a function in particular of the connection and of the status of the router, the arrival of a "2-way Received" event causes the state machine to go either to an "ExStart" state or to a "2-way" state.

The "2-way" state is reached when the type of connection does not need adjacency to be formed between the two routers. This applies, for example, if they are members of a multi-access network and if neither of the two routers is a designated router. This state is a stable state but it can be called into question by the arrival of a "1-way Received" event indicating that the connection between the two routers has encountered a problem, and that the state machine must return to the "Init" state.

The "ExStart" state is reached when it is necessary to put the two routers into adjacency. During this state, the router negotiates with its counterpart in order to determine a master and a slave, by sending DDP messages without data.

Once this negotiation has been performed, a "NegotiationDone" event occurs and the state machine goes to the "Exchange" state.

During this state, the two routers exchange DDP messages.

Once the two routers have exchanged the description of all of their data, the "ExchangeDone" event occurs.

There are then two possibilities:

either both routers have the same data about the network; they are thus in adjacency, and the state machine goes to the "Full" state;

or else there is divergence, and the router that has the least up to date data makes a request (LS Request) to the other router to transmit a data exchange message in order to update its data; for this purpose, the state machine goes to the "Loading" state.

Conventionally, when a router system is re-started, e.g. after a failure, the state machine thus has to re-start in the "Down" state. The other router is then warned of this change, and it can also be subjected to a change of state.

Clearly, a state machine following the path once again from the "Down" state to an end state such as "2-way" or "Full" takes a long time and can generate heavy traffic over the network (exchanging DDP messages, etc.).

In order to minimize the consequences of the router failing or of it being shut down temporarily for maintenance purposes, it is possible to provide for redundancy in the routers: a router on standby becomes active when the active router stops, e.g. after a failure or after an intentional shut down for maintenance.

Such a solution is, in particular, implemented by Cisco, in the Hot Standby Router Protocol (HSRP), which

is described, for example, at the following Internet address:
http://www.cisco.com/warp/public/619/hsrpguidetoc.html

Another redundancy solution is described in IETF's RFC 2338, entitled "Virtual Router Redundancy Protocol".

However, in that solution as well, when a first router in the active state fails and causes the standby second router to take over, the state machines managing the routers that are neighbors of the router in question must re-start in the "Down" state.

That results in the second router being unavailable for a time lapse before it can return to the state in which the first router was before it failed or stopped. Conventionally, that delay is at least 40 seconds and is generally about 1 minute.

The state machines re-starting also suffers from the drawback of causing changes in the states of the neighboring routers.

OBJECTS AND SUMMARY OF THE INVENTION

An object of the invention is to mitigate those drawbacks. To this end, the invention provides a router made up:

firstly of at least two router modules, only one of which is in an active state at any given time, the others being in a standby state; and

secondly of changeover means making it possible to cause one of said other router modules to go from a standby state to an active state when the router module in the active state stops.

The router is connected to neighbor routers, and said router modules have state machines associated with each of said neighbor routers.

In this router, each of said router modules further has:

data storage means which, in the active state, enable it to store data relating to the states of the state machines, when said machines are in stable states; and

data retrieval means for retrieving said data when the router module changes over to the active state.

Thus, by means of the data storage, the router module in the standby state is capable of taking over in the same state as the state in which the router module in the active state was before it stopped or failed.

The changeover from the active router module to the standby module takes place transparently for the other routers of the network, and without generating any delay due to non-availability.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention and its advantages appear more clearly from the following description of an implementation, given with reference to the accompanying figures, in which:

Figure 1 (already described) shows the state machine managing communications associated with a neighboring router;

Figure 2 shows in detail the "Exchange" state of the invention; and

Figure 3 shows a multi-access network made up of three routers.

MORE DETAILED DESCRIPTION

Figure 3 shows that the router S comprises two router modules MR_1 and MR_2. These two router modules perform the same functions as the routers of the state of the art.

However, in addition, the two router modules have means for communicating between each other.

More precisely, each of the router modules MR_1 and MR_2 has data storage means for storing data relating to the states of its state machines, when said state machines are in stable states, and data retrieval means for retrieving said data. As explained below, these data storage means may be means for writing in a shared memory

M, and the data retrieval means may be means for reading from said shared memory M. However, other implementations of the invention are possible, in particular by using a software bus such as a Common Object Request Broker Architecture (CORBA) bus.

At any given time, only one of the two router modules is in the active state, i.e. is performing its router function. The other router module is in the standby state, i.e. it is invisible to the network but it is ready to take over the function of the active router module if said active router module fails or is shut down.

Data storage is implemented only by the router module in the active state, each time the state machine enters a stable state.

These stable states may be the "Down", "Init", "Extart", "2-way", "Exchange", "Full", and "Loading" states, i.e. the states for which the connection is determined.

In an implementation of the invention, the state of the state machine after a DDP message exchange is added to this list.

As mentioned above, once in the "Exchange" state, the router module exchanges DDP messages with its neighbor. This additional state is the state of the state machine when the associated neighboring router has performed an exchange of messages with the same sequence number, i.e. when it has received a DDP message and has sent a DDP message.

This additional state is shown in Figure 2, which shows in detail the "Exchange" state. In this state, the router module sends a DDP message in a send step represented by box 1, and receives a DDP message in a receive step represented by box 2. The two DDP messages have the same sequence number. The order of these two steps depends on the status of the router module, i.e. whether it is master or slave in the exchange.

After the exchange, the data transmitted via the DDP messages is stored in a step represented by box 3. Then, a step referenced 4 consists in testing whether another exchange should be implemented.

The data to be stored contains at least one identifier for identifying the state of the state machine. However, it is possible also to store other data in order to facilitate starting up the standby router module when necessary.

In an implementation of the invention, data relating to the neighboring routers is stored when creating the state machines and when deleting them. This data is the data making it possible to create the state machine in compliance with Section 10 of above-mentioned RFC 2328.

In an implementation of the invention, when the state machine is in the "Full" state, any LS Update message received is stored.

Figure 3 shows a particular implementation using a shared memory.

In this implementation, the router S has two neighbors S_1 and S_2. Therefore, each of the router modules MR_1 and MR_2 has two state machines, one associated with the neighbor S_1 and the other associated with the neighbor S_2.

In the example, it is assumed that the router S_1 was previously elected as the designated router. Thus, the router S must enter into adjacency with the router S_1. Therefore, in the router S, the state machine associated with the router S_1 is in the "Full" state, and the state machine associated with the router S_2 is in the "2-way" state.

Since these are stable states, as defined above, a state identifier is stored by the router module in the active state (e.g. MR_1) in the shared memory M.

When the router module MR_1 becomes non-operational, e.g. when it is shut down for maintenance or when it fails, the router module MR_2 goes from the standby state to the active state.

Whereupon the router module MR_2 can read back firstly the data relating to the states of both state machines, and secondly the data relating to the neighbor routers, as stored when the state machines were created.

Thus, the router module MR_2 knows that it must force the state machine associated with the router S_1 to go to the "Full" state, and that it must force the state machine associated with the router S_2 to go to the "2-way" state.

The router module MR_2 can then take over the function of the router module MR_1 very quickly and also transparently for the other router modules of the network.

Other implementations are naturally accessible to the person skilled in the art. In particular, the two router modules may communicate via inter-process communications means. For example, the inter-process communications means may be a software bus, such as the CORBA software bus in compliance with the specifications of the Object Management Group (OMG).

The data storage step may then be preceded by a data sending step in which the data is sent to the router module on standby, which module then has to store the data so that it can be retrieved when a state transition occurs.